

SAE Informatique

Paternotte Mattéo
Doussineaux Mathis



Dans ce SAE nous avons étudié un robot nommé Moway. On a appris à coder sur son langage de programmation qui est le langage C. Dans le code il y a le <<Jumping the Bootloader>>, il est donné par la marque et contient les infos et les librairies dont le robot a besoin pour fonctionner correctement. Nous avons fait plusieurs programmes.

Programme 1:

Le programme commence par le main. Quand tout est en majuscule ceux-ci est une fonction d'une librairie.

SEN_CONFIG permet de configurer les entrées et sorties nécessaires pour gérer les capteurs et initialiser les variables .

while(1) veut dire que tant qu' on a 1 on répète l'allumage des LED ,l'attendre , éteindre les LED et attendre .

$TCY = 4/FOSC = 4/4*10^{**6} = 4*10^{**-6}$ microseconde

$delay10ktcy = 50*10000*4*10^{**-6} = 0.5s$

```
void main() {  
    SEN_CONFIG(); //SENSOR CONFIGURATION  
    while(1) { //du booléen.Si on a(1),c'est vrai (0)= faux et permet de faire une boucle  
infinie  
        LED_TOP_GREEN_ON();  
        Delay10KTCYx(50);  
        LED_TOP_GREEN_OFF();  
        Delay10KTCYx(50);  
    }  
}
```

Programme 2: gestion du capteur de luminosité

Le programme devras recréer l'algorithme suivant :

début

si la luminosité est insuffisante alors

allumer la LED frontale

fin si

fin

En suivant l'algorithme, le programme devra allumer la LED frontale si la luminosité est insuffisante.

```
void main() {  
    SEN_CONFIG();  
    while(1) {  
        if (SEN_LIGHT() <= 10) {  
            LED_FRONT_ON();  
        }  
        else {  
            LED_FRONT_OFF();  
        }  
    }  
}
```

Programme 3: détection d'obstacle en numérique

Le programme devras recréer l'algorithme suivant :

début

si un obstacle est présent devant ou sur les côtés du robot alors

émettre un bip sonore

fin si

fin

Le programme doit émettre un signal sonore s' il détecte un obstacle.

```
void main() {  
    SEN_CONFIG();  
    while(1) {  
        if (SEN_OBS_DIG(0)==1 || SEN_OBS_DIG(1)==1 || SEN_OBS_DIG(2)==1 ||  
SEN_OBS_DIG(3)==1 ) {  
            SEN_SPEAKER(1000, 2000, 1);  
        }  
        else {  
            SEN_SPEAKER(1000, 500, 0);  
        }  
    }  
}
```

Programme 4: détection d'obstacle en mode alogue

L'algorithme est le même que celui précédent.

Nous devons faire le même programme qu'avant mais les sensors seront en analogue.

```
void main() {  
    SEN_CONFIG();  
    while(1) {  
        if (SEN_OBS_ANALOG(0)>=1 || SEN_OBS_ANALOG(1)>=1 ||  
SEN_OBS_ANALOG(2)>=1 || SEN_OBS_ANALOG(3)>=1 ) {  
            SEN_SPEAKER(2, 2000, 1);  
        }  
        else {  
            SEN_SPEAKER(2, 500, 0);  
        }  
    }  
}
```

Programme 5: ligne droite est demi-tour

Le programme devras recréer l'algorithme suivant :

i : entier

début

attendre 2s

émettre 2 bips sonores

pour i <- 1 à 2 **faire**

faire avancer le robot en ligne droite à 3/4 de sa vitesse maximale

pendant 3s

faire une rotation de 180° à gauche

fin pour

fin

Le programme fait attendre 2 secondes puis doit émettre 2 bips sonores faire avancer le robot en ligne droite à 3/4 de sa vitesse maximale puis pendant 3 secondes faire une rotation de 180° à gauche.

```
unsigned char i;  
SEN_CONFIG();  
MOT_CONFIG();  
Delay1KTCYx(2000);  
SEN_SPEAKER(255, 0, 1);  
Delay1KTCYx(2000);  
SEN_SPEAKER(0, 0, 0);  
Delay1KTCYx(2000);  
SEN_SPEAKER(255, 0, 1);  
Delay10KTCYx(2000);  
SEN_SPEAKER(0, 0, 0);  
for (i=1;i<=2;i++) {  
    MOT_STR(75,FWD,DISTANCE,30);  
    while(!MOT_END);  
    MOT_ROT(100,FWD,CENTER,RIGHT,DISTANCE,180);}
```

Programme 6: trajectoire carrée

Le programme devras recréer l'algorithme suivant :

i : entier

y : entier

début

attendre 2s

émettre 2 bips sonores

pour y <- 0 à 2 faire

pour i <- 0 à 3 faire

 faire avancer le robot en ligne droite à 100% de sa vitesse maximale

 allumer la LED frontale

 pendant 0.4s

 éteindre la LED frontale

 pendant 0.4s

fin pour

 faire tourner le robot de 90°degrés

 allumer la LED frontale

 pendant 0.4s

 éteindre la LED frontale

 pendant 0.4s

fin pour

fin

Le programme doit faire que le robot avance en trajectoire carrée de 30 cm tout en faisant clignoter les LEDs avec une fréquence de 2,5 Hz.

```
unsigned char i;
```

```
unsigned char y;
```

```
SEN_CONFIG();
```

```
MOT_CONFIG();
```

```
for (y=0;y<1;y++){
```

```
    for (i=1;i<=3;i++) {
```

```
        MOT_STR(100,FWD,DISTANCE,100);
```

```
        while(!MOT_END){
```

```
            LED_TOP_GREEN_ON();
```

```
            Delay10KTCYx(40);
```

```
            LED_TOP_GREEN_OFF();
```

```
            Delay10KTCYx(40);
```

```
        }
```

```
    }
```

```
    while(!MOT_END);
```

```
    MOT_ROT(100,FWD,CENTER,RIGHT,DISTANCE,25);
```

```
    LED_TOP_GREEN_ON();
```

```
    Delay10KTCYx(40);
```

```
    LED_TOP_GREEN_OFF();
```

```
    Delay10KTCYx(40);
```

```
    }
```

```
}
```

Programme 7: parcourir l'aire de jeu en évitant les bordures, les obstacles et les autres robots

Le programme devras recréer l'algorithme suivant :

début

attendre 2s

émettre 2 bips sonores

si capteur 1 détecte un objet alors

faire tourner le robot

allumer la LED frontale

pendant 0.4s

fin si

si capteur 0 détecte un objet alors

faire tourner le robot

allumer la LED frontale

pendant 0.4s

fin si

si capteur 2 détecte un objet alors

faire tourner le robot

allumer la LED frontale

pendant 0.4s

fin si

si capteur 3 détecte un objet alors

faire tourner le robot

allumer la LED frontale

pendant 0.4s

fin si

faire avancer le robot

fin

Le programme doit permettre au robot de se déplacer dans l'aire de jeux sans toucher les murs, les obstacles et les autres robots.

```
SEN_CONFIG();
```

```
MOT_CONFIG();
```

```
while(1) {
```

```
    if (SEN_OBS_ANALOG(1)>=200) {
```

```
        MOT_ROT(100,FWD,CENTER,RIGHT,DISTANCE,15);
```

```
        LED_FRONT_ON();
```

```
        while(!MOT_END){}
```

```
    }
```

```
    if(SEN_OBS_ANALOG(0)>=200){
```

```
        MOT_ROT(100,FWD,CENTER,RIGHT,DISTANCE,15);
```

```
        LED_FRONT_ON();
```

```
        while(!MOT_END){}
```

```
    }
```

```
    if(SEN_OBS_ANALOG(2)>=200){
```

```
        MOT_ROT(100,FWD,CENTER,LEFT,DISTANCE,10);
```

```
        LED_FRONT_ON();
```

```
        while(!MOT_END){}
```

```

    }
    if(SEN_OBS_ANALOG(3)>=200){
        MOT_ROT(100,FWD,CENTER,LEFT,DISTANCE,10);
        LED_FRONT_ON();
        while(!MOT_END){}
    }
    MOT_STR(100,FWD,DISTANCE,100);
}
}

```

Programme 8: le petit circuit

Le programme doit nous faire parcourir un circuit dans le moindre temps. Il devra aussi arrêter le robot 2 secondes sur la ligne de départ et s'arrêter définitivement sur la ligne d'arrivée.

```

void go(){
    int x=150;
    SEN_CONFIG();
    MOT_CONFIG();
    LED_FRONT_ON();
    if ((SEN_OBS_ANALOG(0)>x||SEN_OBS_ANALOG(1)>x)){
        MOT_ROT(100,FWD,CENTER,RIGHT,ANGLE,25);
    }
    else if (SEN_OBS_ANALOG(2)>x||SEN_OBS_ANALOG(3)>x){
        MOT_ROT(100,FWD,CENTER,LEFT,ANGLE,18);
    }
    else
        MOT_STR(100,FWD,DISTANCE,0);}

void main() {
    unsigned char y=1;
    SEN_CONFIG();
    MOT_CONFIG();
    while(1) {
        while((SEN_LINE_DIG(0)==0||SEN_LINE_DIG(1)==0)){
            MOT_STR(0,FWD,DISTANCE,0);
            Delay10KTCYx(200);
            MOT_STR(100,FWD,DISTANCE,10);
            y=y+1;
            Delay10KTCYx(20);
        }
        while((SEN_LINE_DIG(0)==1||SEN_LINE_DIG(1)==1) && (y<=2)){
            go();
        }
    }
}

```

Le programme doit nous faire parcourir un circuit dans le moindre temps. Il devra aussi arrêter le robot 2 secondes sur la ligne de départ et s'arrêter définitivement sur la ligne d'arrivée.

```

void ah(){
    if(SEN_OBS_ANALOG(OBS_SIDE_L)>200){
        MOT_ROT(80,FWD,CENTER,RIGHT,ANGLE,5);
    }
    if(SEN_OBS_ANALOG(OBS_SIDE_R)>200){
        MOT_ROT(80,FWD,CENTER,LEFT,ANGLE,5);
    }
}

void main(){
    int q=0;
    MOT_CONFIG();
    SEN_CONFIG();
    while(a==1){
        if(q==0){
            if(!(SEN_LINE_DIG(0)==1||SEN_LINE_DIG(1)==1)) {
                MOT_STOP();
                Delay10KTCYx(100);
                MOT_STR(80,FWD,DISTANCE,100);
                while(!MOT_END);
            }
        }
        if(q>10){
            if(!(SEN_LINE_DIG(0)==1||SEN_LINE_DIG(1)==1)) {
                MOT_STOP();
                Delay10KTCYx(100);
                while(!MOT_END);
                while(1);
            }
        }
        MOT_STR(80,FWD,DISTANCE,0);
        ah();
    }
}

if(SEN_OBS_ANALOG(OBS_CENTER_L)>240||SEN_OBS_ANALOG(OBS_CENTER_R)>240) {
    q++;
    switch(q){
        case 1 :
            MOT_ROT(80,FWD,CENTER,RIGHT,ANGLE,21);
            while(!MOT_END);
            break;
    }
}

```

```
case 2 :  
MOT_ROT(80,FWD,CENTER,RIGHT,ANGLE,25);  
while(!MOT_END);  
break;
```

```
case 3 :  
MOT_ROT(80,FWD,CENTER,LEFT,ANGLE,25);  
while(!MOT_END);  
break;
```

```
case 4 :  
MOT_ROT(80,FWD,CENTER,LEFT,ANGLE,31);  
while(!MOT_END);  
break;
```

```
case 5 :  
MOT_STR(100,BACK,DISTANCE,31);  
while(!MOT_END);  
MOT_ROT(80,FWD,CENTER,RIGHT,ANGLE,25);  
while(!MOT_END);  
MOT_STR(100,FWD,DISTANCE,255);  
while(!MOT_END);  
MOT_ROT(80,FWD,CENTER,RIGHT,ANGLE,25);  
while(!MOT_END);  
MOT_STR(80,FWD,DISTANCE,175);  
while(!MOT_END);  
MOT_ROT(80,FWD,CENTER,LEFT,ANGLE,26);  
while(!MOT_END);  
break;
```

```
case 6 :  
MOT_ROT(80,FWD,CENTER,RIGHT,ANGLE,24);  
while(!MOT_END);  
break;
```

```
case 7 :  
MOT_ROT(80,FWD,CENTER,RIGHT,ANGLE,26);  
while(!MOT_END);  
break;
```

```
case 8 :  
MOT_STR(100,BACK,DISTANCE,49);  
while(!MOT_END);  
MOT_ROT(80,FWD,CENTER,RIGHT,ANGLE,24);  
while(!MOT_END);  
break;
```



```

case 9 :
MOT_STR(100,BACK,DISTANCE,50);
while(!MOT_END);
MOT_ROT(80,FWD,CENTER,RIGHT,ANGLE,26);
while(!MOT_END);
break;

case 10 :
MOT_STR(100,FWD,DISTANCE,12);
while(!MOT_END);
MOT_ROT(80,FWD,WHEEL,LEFT,ANGLE,25);
while(!MOT_END);
break;

case 11 :
MOT_ROT(80,FWD,CENTER,LEFT,ANGLE,25);
while(!MOT_END);
MOT_STR(100,FWD,DISTANCE,150);
while(!MOT_END);
MOT_ROT(80,FWD,CENTER,LEFT,ANGLE,25);
while(!MOT_END);
MOT_STR(100,FWD,DISTANCE,50);
while(!MOT_END);
MOT_ROT(80,FWD,CENTER,RIGHT,ANGLE,25);
while(!MOT_END);
break;

default :
break;
}
}
}
}
}

```

Grâce au SAE j'ai pu approfondir ma connaissance sur le langage de programmation et j'ai pu voir en temps réel ce que je codé. J'ai aussi appris comment chercher ce que je voulais dans les document ressources.